

Multiple Electron Scattering Routines for PEREGRINE

J.A. White

August 23, 1999

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Work performed under the auspices of the U. S. Department of Energy by the University of California Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (423) 576-8401
<http://apollo.osti.gov/bridge/>

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161
<http://www.ntis.gov/>

OR

Lawrence Livermore National Laboratory
Technical Information Department's Digital Library
<http://www.llnl.gov/tid/Library.html>

Multiple electron scattering routines for PEREGRINE

Jody A. White, LLNL

August 23, 1999

1 Overview

The Monte Carlo electron scattering routines solve multiple elastic scatters in a condensed history approach. The Goudsmit-Saunderson scattering model is used and its implementation is taken from Kawrakow and Bielajew[1]. The subroutines produce an exit angle representing a likely scattering angle of a single incident electron after scattering elastically over a given step size. Two input parameters, λ and η , that depend on the atomic species and incident energy must first be specified. The mapping from species and energy to η and λ already existed in the PEREGRINE code and was not redone or modified in any way. The software has been validated by comparisons to Moliere and Goudsmit-Saunderson models of D.W.O. Rogers[2].

As required by licensing considerations, no public domain or copyrighted software has been used in any phase of the preparation of any of these subroutines or data files. Apart from needing to have η and λ specified through PEREGRINE, the code provided is completely self-contained. Everything is written in the FORTRAN 77 language to simplify inclusion in the existing PEREGRINE package.

2 How things work

Several data files are read first. These files describe a grid of $q^{2+}(u)$ points calculated at different λ and η values to allow on-the-fly interpolation of scattering amplitudes in PEREGRINE. The grid spacing is the same as specified by Kawrakow and Bielajew. Variables *nueta* and *nulambda* are the two numerical inputs.¹ Subroutine *getangle* returns *costheta* as the desired scattering parameter. As one would expect, *costheta* is actually the cosine of the scattering angle.

Subroutine *getangle* first determines from λ whether the type of scatter will be a no scatter, a single scatter, or a two+ scatter. These expressions are also described in Kawrakow and Bielajew. If no scatter is selected, the output cosine is 1.0 and the routine ends. If a single scatter is selected, the final scatter angle is obtained by sampling from the single scatter amplitude calculated as a function of $\cos(\phi)$. Single scattering is assumed to have a screened Rutherford form. The Rutherford cross section is computed in the function *cross*. If the two+ scatter process is selected, the scatter angle is naturally found by sampling from the two+ scattering amplitude. The two+ scatter amplitude itself is determined by interpolation from the input data files described earlier.

¹Note that *getangle* uses variables *lambda* and *eta* as well as *nulambda* and *nueta*. The former are two-dimensional array variables describing the grid of λ and η values over which $q^{2+}(u)$ exists.

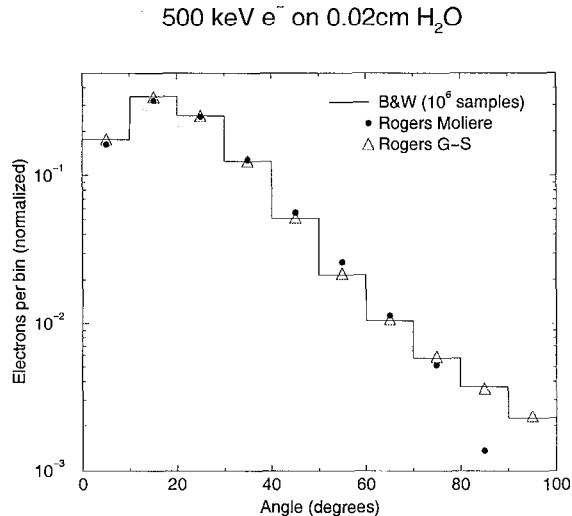


Figure 1: Scatter results for 500 keV e^- on 0.02cm H_2O . B&W indicates the author's method. The Rogers results are referenced in the text.

The *interp* subroutine performs the actual interpolation. Newton's forward difference method is the chosen interpolation scheme. It is described in Abramowitz and Stegun[3].

3 Results

Results for this Goudsmit-Saunderson implementation are shown in Fig. 1 as the curve labelled B&W for the scattering of 500 keV electrons through 0.02 cm of water. Rogers has provided modeling results of his Moliere and Goudsmit-Saunderson (G-S) curves for comparison. The agreement between the present implementation of Goudsmit-Saunderson scattering and that of Rogers is excellent. The Moliere model results deviate at higher angles ($\theta \geq 50$ degrees), as would be expected.

4 Inclusion in PEREGRINE

To include the subroutines in PEREGRINE, the main program must read the data files referenced in *driver.f* and pass their contents to *getangle*. The data files are currently in ascii format, but could of course be stored in binary format instead. The required data files are *lambda.dat*, *eta.dat*, *worlda.dat*, *worldq.dat*, *worldu.dat* and *worldcosx.dat*. The must-have code components are *getangle*, *interp*, and the function *cross*.

A function called *ran3* was used originally in *getangle* as a random number generator with output range from 0 to 1. This is not the same one that exists in the PEREGRINE code (*ranf4*), so be sure you have the proper random number generator. Do NOT mess with anything else in *getangle* or, especially, *interp*!

5 Inclusion in other transport packages

Usage is similar as for PEREGRINE, but appropriate mappings from energy and species to η and λ must be provided. *Interp* is a bottleneck. The code will

run much faster if you incorporate the routines so that calls to *interp* are not repeated for incident particles with the same energy.

6 Producing the data files

J. White has retained all the subroutines that were used to calculate the exact $q^{2+}(u)$ surfaces. Only the final output q^{2+} -surfaces are required in PEREGRINE. The reason for only reading in the output files within PEREGRINE rather than calculating everything from scratch during the initialization period of each run is that the surface calculations can drain cpu. (This is why Kawrakow and Bielajew devised their interpolation scheme in the first place.)

The modified Bessel function $K_1(x)$ required to calculate the $q^{2+}(u)$ surfaces was determined from polynomial approximations given in Abramowitz and Stegun[3]. The Legendre polynomials were found from simple recursion.

7 Acknowledgement

I acknowledge Paul Bergstrom for his help getting started and incorporating the routines into Peregrine for testing. I also acknowledge Christine Hartmann-Siantar for enthusiastically recruiting me to the project.

References

- [1] Kawrakow, I. and Bielajew, A.F. *Nucl Instr and Meth.*, **B134**, 325 (1998).
- [2] D.W.O. Rogers. *Nucl Instr and Meth.*, **227** 535 (1984).
- [3] M. Abramowitz and I.E. Stegun, editors. Handbook of Mathematical Functions, National Bureau of Standards Applied Mathematics Series, 1964.